


Amendments to the Specification

Please replace paragraph 4 on page 5 with the following amended paragraph:

- A
- The CLPs concurrently execute code that is stored in a common Instruction Memory. Each CLP includes a core and a 3-stage pipeline, 16 GPRs (general purpose registers) and an ALU (arithmetic logic unit). The coprocessors can execute operation concurrently with each other and concurrently with the CLP. When the coprocessor is interfaced with the CLP, it extends the basic CLP instruction and register model. The commands of each coprocessor appear as new assembler mnemonics, and the registers of the coprocessor appear to the CLP programmer as new scalar and array registers. Some of the array registers are located in the shared memory pool (44) (shown as 44 in Figure 1). The coprocessor can execute asynchronously with the CLP. This allows the CLP to continue processing instructions while a coprocessor is executing a command. A wait instruction causes the CLP to wait until a coprocessor execution command is completed.

Please replace paragraph 2 on page 18 with the following amended paragraph:

The execution interface (16) enables the CLP (34) to initiate command execution on any of the coprocessors (20, 22...). The coprocessor number (411) selects one of 16 coprocessors as the target for the command. When the CLP activates the start field (410) to logical 1, the selected coprocessor ~~(450)~~ 411 as indicated by coprocessor number (411) begins executing the command specified by the 6-bit Op field (412). The op arguments (413) are 44 bits of data that are passed along with the command for the coprocessor ~~(450)~~ 411 to process. The busy signal (414) is a sixteen-bit field, one bit for each coprocessor (401), and indicates whether a coprocessor is busy executing a command (bit = 1) or whether that coprocessor is not executing a command (bit = 0). These 16 bits are stored in the scalar register where bit 0 of the register corresponds to coprocessor 0, bit 1 to coprocessor 1, etc. The OK/K.O. field (415) is a sixteen-bit field, one bit for each coprocessor (401). It is a one-bit return value code which is command specific. For example, it may be used to indicate to the CLP (34) whether a command given to a coprocessor (401) ended with a failure, or whether a command was successful. This information is stored within the CLP scalar register where bit 0 of the register corresponds to coprocessor 0, bit 1 to coprocessor 1, etc. The direct/indirect field (417) indicates to the selected coprocessor ~~(450)~~ 411 which format of the Coprocessor Execute instruction is executing. If direct/indirect = 0, then direct format is executing. else if direct/indirect = 1 then the indirect format is executing.

Please replace paragraphs 1 and 2 on page 19 with the following amended paragraphs:

A3
The write interface uses the write field (419) to select a coprocessor ~~(450)~~ 411 indicated by the coprocessor number (420). The write field (419) is forced to one whenever the CLP (34) wants to write data to the selected coprocessor. The coprocessor register identifier (421) indicates the register that the CLP (34) will write to within the selected coprocessor ~~(450)~~ 411. The coprocessor register identifier (421) is an eight-bit field and accordingly (256) registers are supported. A coprocessor register identifier the range 0 to 239 indicates a write to a scalar register. A coprocessor register identifier in the range 240 to 255 indicates a write to an array register. In the case of an array register write, the offset field (422) indicates the starting point for the data write operation in the array register. This field is eight-bits in size and therefore will support 256 addresses within an array. The data out field (423) carries the data that will be written to the coprocessor ~~(450)~~ 411. It is 128 bits in size, and therefore up to 128 bits of information may be written in one time. The write valid field (424) indicates to the CLP (34) when the coprocessor ~~(450)~~ 411 is finished receiving the data. This allows the CLP (34) to pause and hold the data valid while the coprocessor (450) takes the data.

The read interface (14) is similar in structure to the write interface (16) except that data is read from the coprocessor. The read field (428) corresponds to the write field (419), and is used by the CLP (34) to indicate when a read operation is to be performed on the selected coprocessor ~~(450)~~ 411. The coprocessor number identifier field (429) determines which coprocessor ~~(450)~~ 411 is selected. The register number field (430), offset field (431), and read valid field (433) correspond to (421) (422), and (424) in the write interface. The data-in field (432) carries the data from the coprocessor ~~(450)~~ 411 to the CLP (34).